Discrete Applied Mathematics 161 (2013) 377–394

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Stability measure for a generalized assembly line balancing problem

Evgeny Gurevsky *, Olga Battaïa, Alexandre Dolgui

École Nationale Supérieure des Mines de Saint-Étienne, 158, cours Fauriel, 42023 Saint-Étienne Cédex 2, France

ARTICLE INFO

Article history:
Received 1 March 2011
Received in revised form 6 June 2012
Accepted 31 August 2012
Available online 6 October 2012

Keywords:
Stability/Sensitivity analysis
Uncertainty
Line balancing

ABSTRACT

A generalized formulation for assembly line balancing problem (GALBP) is considered, where several workplaces are associated with each workstation. Thus, all tasks assigned to the same workstation have to be partitioned into blocks: each block regroups all tasks to be performed at the same workplace. The product items visit all workplaces sequentially, therefore, all blocks are proceeded in a sequential way. However, the tasks grouped into the same block are executed simultaneously. As a consequence, the execution of a block takes only the time of its longest task. This parallel execution modifies the manner to take into account the cycle time constraint. Precedence and exclusion constraints also exist for workstations and their workplaces. The objective is to assign all given tasks to workstations and workplaces while minimizing the line cost estimated as a weighted sum of the number of workstations and workplaces. The goal of this article is to propose a stability measure for feasible and optimal solutions of this problem with regard to possible variations of the processing time of certain tasks. A heuristic procedure providing a compromise between the objective function and the suggested stability measure is developed and evaluated on benchmark data sets.

1. Introduction

The design of a typical flow-oriented paced production line is considered. The line consists of a number of workstations aligned serially along a conveyor belt. Identical product items are consequently launched down the line and processed at every workstation in the order of their location. A workstation deals with only one product item at a time. The items are transferred from their current workstation to the next one at the end of each time interval called *line cycle time*. All workstations function simultaneously performing elementary tasks assigned to them. Tasks can be executed by a human operator or using special automatic machines installed at workstations.

The design aim is to partition the given set of all elementary tasks into workstations while respecting existing technological and economical constraints and optimizing one or several objectives. The set of tasks assigned to a workstation determines its load. The working time of a workstation on a product item must not be greater than the cycle time. A workstation with the greatest working time is called the *most loaded*.

This optimization problem is one of the important issues of managing assembly lines. Its simple version, the simple assembly line balancing problem or SALBP, takes into account only precedence and cycle time constraints where the sum of tasks assigned to the same workstation must be not greater than the cycle time. With regard to objectives employed, SALBPs are commonly classified into three types [25,33]: minimize the total number of opened workstations for a fixed line cycle time (SALBP-1); minimize the working time on the most loaded workstation with a fixed number of workstations (SALBP-2); and if neither the number of workstations nor line cycle time is fixed, maximize the *line efficiency* (SALBP-E). The latter objective minimizes the number of opened workstations $\times$ working time on the most loaded one. It should be emphasized that all these problems are known to be $\mathcal{NP}$-hard [26, Chapter 2.2.1.5].

* Corresponding author. Fax: +33 477426666.
E-mail addresses: evgeny.gurevsky@gmail.com (E. Gurevsky), battaia@emse.fr (O. Battaïa), dolgui@emse.fr (A. Dolgui).

0166-218X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2012.08.037

In this paper, a generalization of SALBP-1 is considered. Namely, it is supposed that each workstation is equipped with one or several workplaces (blocks) activated sequentially. At the same time, the tasks assigned to the same workplace (block) are executed simultaneously. Therefore, the working time on a workstation is determined as the sum of the working time of blocks belonging to this workstation, while the working time of a block is determined as the maximal processing time among the tasks assigned to it. The goal is to minimize the number of used equipment (i.e. the total number of workstations and workplaces). Several industrial examples of such lines can be found in [2,10,15–17] where blocks correspond to multi-spindle heads and workstations to unit-built machines.

In SALBPs, all task processing times were considered deterministic. However, these times may vary during the line lifecycle because of multiple factors, such as: operator skill, motivation and fatigue, changes in material composition of product items, product and workstation characteristics, etc. To take into account the variability of processing times, the following models are often used in the literature: *stochastic processing times* [1,3,9,12,13,23,35] and *fuzzy processing times* [14,20,34].

For stochastic models, task processing times are commonly assumed to be normally distributed independent random variables with known means and variances. In this case, chance constraints can be introduced. These constraints assure that the probability of the respect of the cycle time for each workstation will be greater than a pre-determined confidence level that is usually equal to 0.95. In fuzzy models, task processing times are represented by fuzzy intervals with given membership functions (possibility distributions) giving the grade of satisfaction of a decision maker. In that case, the assignment of tasks to workstations is implemented with respect to an introduced fuzzy arithmetic.

However, it should be noted that the application of these two models in practice is a difficult task. Indeed, available knowledge on input data is not always sufficient to deduct appropriate probability or possibility distribution functions for task processing times, especially if the design of an assembly line is planed just for one time. More often, a decision maker can only indicate a subset of tasks which processing times are subject to frequent variations. In such cases, another model can be suggested, where the set of given tasks is divided into 2 subsets of constant and variable tasks. This approach was used by Sotskov et al. [28] for SALBP-1. The authors studied the influence of variations of task processing times (VTPT) on optimal solutions constructed for completely deterministic problem. The principal goal of this approach is to determine the limit level of independent VTPT (named the *stability radius*) under which a solution remains optimal. The stability radius is an appropriate measure of credibility of known solutions in presence of VTPT. If the stability radius is known, then will be no need to reconstruct an optimal solution if the VTPT observed do not exceed it.

Note that similar approaches have been already studied for different types of combinatorial optimization problems, where along with the stability radius, another measure of sensitivity called sensitivity interval (the interval of one parameter where the solution preserves its optimality) was investigated. In what follows, we present a short review of these approaches.

Belgacem, Hifi et al. studied the sensitivity of an optimal solution for knapsack and sharing knapsack problems [4–7,18,19] subject to perturbations of profits and weights of the problem. The authors proposed algorithms for calculating the sensitivity intervals for these parameters or, as it was done in [19], while seeking an optimal solution, they adapted a branch and bound technique for calculating this interval.

In [22,36], the authors studied different aspects of sensitivity for the salesman problem. In particular, they considered the problem of seeking k best solutions under the condition that an optimal solution and its stability radius are known. A polynomial algorithm for this problem was presented for $k = 2$, and it was proved that it is $\mathcal{NP}$-hard for $k > 2$.

In [24], the authors considered the shortest path problem for the undirected graphs with m edges. They proved that the sensitivity interval for the length of an edge can be calculated in $\mathcal{O}(m + k \log k)$, where k is the number of edges of the optimal path studied.

Bräsel et al. [8], Kravchenko et al. [21], Sotskov [27] and Sotskov et al. [29–32] study the stability radius of an optimal solution in scheduling under job time uncertainty. Their works are applied on the large range of scheduling problems essentially for job shop and open shop types. They presented the necessary and sufficient conditions for the existence of the strictly positive stability radius as well as the formula of its calculation.

In this paper, we study the stability aspects for both feasible and optimal solutions for a generalized assembly line balancing problem with workplaces of parallel tasks. The remainder of the paper is organized as follows. In Section 2, basic definitions and properties are introduced. Sections 3–5 are devoted to the calculation of the stability radius for feasible, quasi-feasible (see the definition in Section 2), and optimal solutions, respectively. A heuristic procedure to find a compromise between the values of the objective function and the stability radius of a feasible solution is described in Section 6. Experimental results carried out on industrial case benchmarks are analyzed in Section 6.3. Final remarks and conclusions are given in Section 7.

2. Basic definitions and properties

2.1. Feasible, quasi-feasible and optimal solutions

All elementary tasks required to be performed constitute a given set $V = \{1, 2, \ldots, n\}$ associated with a vector $t = (t_1, t_2, \ldots, t_n) \in \mathbb{R}^n_+$ of processing times, where t_j is the processing time of task $j \in V$ and $\mathbb{R}_+$ is the set of all positive real numbers. In this paper, we consider that set V contains two types of tasks:

- Uncertain tasks: their processing times can vary during the line life cycle. The set of such tasks is denoted by $\widetilde{V}$.
- Constant tasks: their processing times remain the same during the line life cycle. Such tasks constitute set $V \setminus \widetilde{V}$.

Without loss of generality, we suppose that $\widetilde{V} = \{1, 2, \dots, \tilde{n}\}$ and $V \setminus \widetilde{V} = \{\tilde{n}+1, \tilde{n}+2, \dots, n\}$, where $0 < \tilde{n} \le n$.

VTPT from set V can be represented by vector $\xi = (\xi_1, \xi_2, \dots, \xi_{\tilde{n}}, 0, 0, \dots, 0) \in \mathbb{R}^n$, where $\xi_j, j \in \widetilde{V}$, can be both positive or negative. Thus, the vector of task processing times in a certain moment of the line life can be represented by *perturbed* vector $t^* = (t_1 + \xi_1, t_2 + \xi_2, \dots, t_{\tilde{n}} + \xi_{\tilde{n}}, t_{\tilde{n}+1}, \dots, t_n)$.

Remark 1. In this study, it is supposed that $t_j^* = \max\{0, t_j + \xi_j\}, j \in \widetilde{V}$.

The set of tasks must be assigned to blocks and these blocks to workstations taking into account the following restrictions:

- *Capacity constraints* are characterized by the line cycle time T_0 and maximal number b_0 of blocks per workstation.
- *Exclusion constraints* define the groups of tasks that cannot be assigned to the same workstation (block) because of their technological incompatibility. These constraints are represented by family E^w (E^b) of pairs of V such that all elements of the same pair $e \in E^w$ ($e \in E^b$) cannot be assigned to the same workstation (block).
- *Precedence constraints* define non-strict partial order relations among tasks and are given by an acyclic direct graph $G = (V, \mathcal{A})$. An arc (i, j) belongs to $\mathcal{A}$ iff task j is assigned to a block that does not precede the block where task i is assigned. However, tasks i and j can be assigned to the same block.

Definition 1. An assignment of tasks V to blocks and of these blocks to workstations is called a feasible solution if no capacity, exclusion, or precedence constraint is violated.

Hereafter, the set of feasible solutions for a given vector $t \in \mathbb{R}^n_+$ is denoted as $\mathbf{S}_{\mathcal{F}}(t)$, where each solution s is characterized by the following collection $\{\{V_{11}, V_{12}, \dots, V_{1|B_1|}\}, \dots, \{V_{|W^s|1}, V_{|W^s|2}, \dots, V_{|W^s|,|B_{|W^s|}|}\}\}$ of non-intersecting nonempty subsets of V such that $V = \bigcup_{k \in W^s} \bigcup_{l \in B_k} V_{kl}$. Here

- W^s is the set of workstations,
- $B_k, k \in W^s$, is the set of blocks of workstation k ($|B_k| \le b_0$),
- $V_{kl}, l \in B_k, k \in W^s$, is the set of tasks assigned to block l of workstation k.

Using the notations introduced, the capacity constraints can be expressed as follows:

$$t_w(V_k) := \sum_{l \in B_k} t_b(V_{kl}) \le T_0, \quad k \in W^s,$$

$$t_b(V_{kl}) := \max\{t_j : j \in V_{kl}\},$$

where $V_k = \bigcup_{l \in B_k} V_{kl}$ is the set of tasks assigned to workstation k, $t_w(V_k)$ is the working time of workstation k, $t_b(V_{kl})$ is the working time of block l of workstation k.

The objective function can be represented as follows:

$$\mathcal{Z}(s, t) := C_1|W^s| + C_2 \sum_{k \in W^s} |B_k| \to \min_{s \in \mathbf{S}_{\mathcal{F}}(t)},$$

where C_1 and C_2 are the costs of one workstation and one block, respectively.

Definition 2. A feasible solution with the minimum value of $\mathcal{Z}(s, t)$ is called optimal solution. The set of optimal solutions is denoted $\mathbf{S}_{\mathcal{O}}(t)$. Obviously, $\mathbf{S}_{\mathcal{O}}(t) \subseteq \mathbf{S}_{\mathcal{F}}(t)$.

Definition 3. Solution s^0 that respects the precedence and exclusion constraints, and has a workstation $k \in W^{s^0}$ such that $t_w(V_k) > T_0$ is called quasi-feasible solution. The set of such solutions is denoted by $\mathbf{S}_{\widehat{\mathcal{F}}}(t)$.

2.2. Stability measure

Note that VTPT modify neither precedence nor exclusion constraints and do not change the number of workstations or blocks. Nevertheless, the feasibility of a solution can be lost, if the working time on the most loaded workstation becomes greater than T_0 for a new perturbed vector t^*. An optimal solution s found for original vector t may lose its optimality for some new perturbed vector t^*, if there is a solution $s^0 \in \mathbf{S}_{\mathcal{F}}(t^*)$ such that $\mathcal{Z}(s^0, t^*) < \mathcal{Z}(s, t^*)$.

Note also that a quasi-feasible solution s^0 that respects the precedence and exclusion constraints, and has a workstation $k \in W^{s^0}$ such that $t_w(V_k) > T_0$ can become feasible and even optimal for a new perturbed vector t^*, if $t^*_w(V_k) \le T_0$ holds for any $k \in W^{s^0}$.

Because of aforementioned perturbations, we have to study the stability of feasible, quasi-feasible and optimal solutions under VTPT. Correspondingly, $\mathcal{R} \in \{\mathcal{F}, \widehat{\mathcal{F}}, \mathcal{O}\}$-stability radii are considered, where $\mathcal{F}$, $\widehat{\mathcal{F}}$ and $\mathcal{O}$ designate feasibility, quasi-feasibility and optimality, respectively.

In order to model VTPT, the Chebychev distance between two vectors t and t' from $\mathbb{R}^n_+$ is used:

$$\left\| t - t' \right\| = \max\{|t_i - t'_i| : i \in V\}.$$

This induces the notion of ε-*neighborhood* of t over $\mathbb{R}^n_+$:

$$\Omega(\varepsilon, t) = \left\{ t' \in \Psi(t) : \left\| t - t' \right\| < \varepsilon \right\}, \quad \varepsilon > 0,$$

where

$$\Psi(t) = \{t' \in \mathbb{R}^n_+ : t'_j = t_j, j \in V \setminus \widetilde{V}\}.$$

Definition 4. Solution $s \in \mathbf{S}_{\mathcal{R}}(t)$ is called $\mathcal{R}$-stable if there exists an ε-neighborhood $\Omega(\varepsilon, t)$ such that for any $t' \in \Omega(\varepsilon, t)$, s remains in $\mathbf{S}_{\mathcal{R}}(t')$, i.e. s is $\mathcal{R}$-stable if the following condition holds:

$$\Xi_{\mathcal{R}}(s, t) = \{\varepsilon > 0 : \forall t' \in \Omega(\varepsilon, t)\ (s \in \mathbf{S}_{\mathcal{R}}(t'))\} \neq \emptyset.$$

Definition 5. $\mathcal{R}$-stability radius $\rho_{\mathcal{R}}(s, t)$ of solution $s \in \mathbf{S}_{\mathcal{R}}(t)$ is defined as the least upper limit of $\Xi_{\mathcal{R}}(s, t)$, i.e.

$$\rho_{\mathcal{R}}(s, t) = \sup \Xi_{\mathcal{R}}(s, t).$$

Remark 2. In this study, it is supposed that $\sup \emptyset = 0$, $\max \emptyset = 0$, $\inf \emptyset = +\infty$, $\min \emptyset = +\infty$.

The $\mathcal{R}$-stability radius of solution $s \in \mathbf{S}_{\mathcal{R}}(t)$ can be considered as the maximal radius of an opened ball over $(\mathbb{R}^n_+, \|\cdot\|)$ with the center at point t such that s remains in $\mathbf{S}_{\mathcal{R}}(t')$ whatever a perturbed vector t' within this ball.

It can be seen that s is $\mathcal{R}$-stable (not $\mathcal{R}$-stable) iff $\rho_{\mathcal{R}}(s, t) > 0$ ($\rho_{\mathcal{R}}(s, t) = 0$); and $\rho_{\mathcal{O}}(s, t) \leq \rho_{\mathcal{F}}(s, t)$ holds for any optimal solution s.

In forthcoming sections, the complexity of the calculation of $\mathcal{R}$-stability radius is evaluated for feasible, quasi-feasible and optimal solutions, respectively.

Hereafter the following property is used.

Property 1. *For any solution s, the following is true:*

$$\forall \varepsilon > 0\, \forall t' \in \Omega(\varepsilon, t)\, \forall k \in \widetilde{W}^s\, \forall l \in \widetilde{B}_k \quad \left(t_b(\widetilde{V}_{kl}) - \varepsilon < t'_b(\widetilde{V}_{kl}) < t_b(\widetilde{V}_{kl}) + \varepsilon \right).$$

Here $\widetilde{W}^s = \{k \in W^s : \widetilde{V}_k \neq \emptyset\}$, $\widetilde{V}_k = V_k \cap \widetilde{V}$, $\widetilde{B}_k = \{l \in B_k : \widetilde{V}_{kl} \neq \emptyset\}$, $\widetilde{V}_{kl} = V_{kl} \cap \widetilde{V}$, $t'_b(V_{kl}) = \max\{t'_j : j \in V_{kl}\}$.

In the next section, the behavior of feasible solutions under VTPT is examined.

3. Stability analysis for feasible solutions

Recall that solution $s \in \mathbf{S}_{\mathcal{F}}(t)$ can lose its feasibility for a new perturbed vector $t^* \in \mathbb{R}^n_+$, i.e. $s \notin \mathbf{S}_{\mathcal{F}}(t^*)$, only if the working time on some workstation becomes greater than T_0, i.e. there exist $k \in \widetilde{W}^s$ such that $t^*_w(V_k) > T_0$.

For each $s \in \mathbf{S}_{\mathcal{F}}(t)$ and $k \in \widetilde{W}^s$, the following notations will be used:

- $\Delta_{kl} = t_b(V_{kl}) - t_b(\widetilde{V}_{kl})$, $l \in \widetilde{B}_k$, is the difference between the maximal processing time of all tasks assigned to l-th block and the maximal processing time of its uncertain tasks,
- $\widetilde{B}_{k0} = \{l \in \widetilde{B}_k : \Delta_{kl} = 0\}$ is the set of blocks where $\Delta_{kl} = 0$, i.e. the maximal task processing time belongs to an uncertain task,
- $\widetilde{B}_{k>} = \{l \in \widetilde{B}_k : \Delta_{kl} > 0\}$ is the set of blocks where $\Delta_{kl} > 0$, i.e. the maximal task processing time belongs to a constant task.

It is not difficult to see that $\widetilde{B}_k = \widetilde{B}_{k0} \cup \widetilde{B}_{k>} \neq \emptyset$, $\widetilde{B}_{k0} \cap \widetilde{B}_{k>} = \emptyset$.

An illustrative example is given in Fig. 1, where a block with four tasks is shown. Each task is represented by a horizontal column which length corresponds to its processing time. Uncertain tasks are cross-hatched.

Lemma 1. *If for $s \in \mathbf{S}_{\mathcal{F}}(t)$ there exists $k \in \widetilde{W}^s$ so that $\widetilde{B}_{k>} \neq \emptyset$, then*

$$\forall l \in \widetilde{B}_{k>}\ \forall t' \in \Omega(\Delta_{kl}, t) \quad (t'_b(V_{kl}) = t_b(V_{kl})), \tag{1}$$

$$\forall l \in \widetilde{B}_{k>}\ \forall \varepsilon \geq \Delta_{kl}\ \forall t' \in \Omega(\varepsilon, t) \quad (t'_b(V_{kl}) \leq t_b(V_{kl}) + \varepsilon - \Delta_{kl}). \tag{2}$$

Proof. By definition

$$t'_b(V_{kl}) = \max\{t'_b(\widetilde{V}_{kl}), t_b(V_{kl})\},$$

and due to Property 1 we have

$$t'_b(\widetilde{V}_{kl}) < t_b(\widetilde{V}_{kl}) + \Delta_{kl} = t_b(V_{kl}), \quad t' \in \Omega(\Delta_{kl}, t),$$

$$t'_b(\widetilde{V}_{kl}) < t_b(\widetilde{V}_{kl}) + \varepsilon = t_b(V_{kl}) + \varepsilon - \Delta_{kl}, \quad t' \in \Omega(\varepsilon, t),$$

this implies the validity of (1) and (2). □

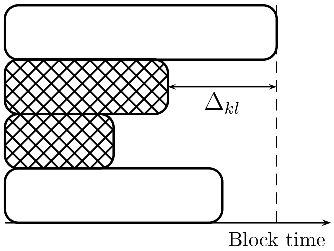

Fig. 1. Block l of workstation k of solution s.

Theorem 1 shows that the $\mathcal{F}$-stability radius calculation can be reduced to a bilevel programming problem. The following notation will be used:

$$\mathcal{I}_n = \begin{cases} \{1, \dots, n\}, & \text{if } n \in \mathbb{N}, \\ \emptyset, & \text{if } n = 0. \end{cases}$$

Theorem 1. *The $\mathcal{F}$-stability radius of $s \in \mathbf{S}_{\mathcal{F}}(t)$ is calculated as follows:*

$$\rho_{\mathcal{F}}(s, t) = \min_{k \in \widetilde{W}^s} \varphi(k), \tag{3}$$

where

$$\varphi(k) = \Delta_{kl_{j_k^*}} + \frac{T_0 - t_w(V_k) - \Theta(j_k^*)}{|\widetilde{B}_{k0}| + j_k^*},$$
$$j_k^* = \operatorname{argmax}\{\Theta(j) : \Theta(j) \le T_0 - t_w(V_k), j \in \{0\} \cup \mathcal{I}_{|\widetilde{B}_{k>}|}\},$$
$$\Theta(j) = (|\widetilde{B}_{k0}| + j)\Delta_{kl_j} - \sum_{i \in \mathcal{I}_j} \Delta_{kl_i},$$
$$0 = \Delta_{kl_0} \le \cdots \le \Delta_{kl_{|\widetilde{B}_{k>}|}}, \qquad l_i \in \widetilde{B}_{k>}, \quad i \in \mathcal{I}_{|\widetilde{B}_{k>}|}.$$

Proof. To simplify the further statement, the following notation is introduced: ρ and φ are the left-hand and the right-hand sides of (3), respectively. Note that φ is a nonnegative finite number due to inclusion $s \in \mathbf{S}_{\mathcal{F}}(t)$ and the definition of j_k^* that also implies the following inequalities

$$|\widetilde{B}_{k0}| + j_k^* > 0, \quad k \in \widetilde{W}^s. \tag{4}$$

Remark that nondecreasing function $\Theta(\cdot)$ has a useful recursive representation:

$$\Theta(j+1) = \Theta(j) + (|\widetilde{B}_{k0}| + j)(\Delta_{kl_{j+1}} - \Delta_{kl_j}), \qquad \Theta(0) = 0, \tag{5}$$

that will be used in the sequel.

To prove formula (3), we consequently show that inequalities $\rho \ge \varphi$ and $\rho \le \varphi$ hold.

First let us prove that $\rho \ge \varphi$. To do this, it is sufficient to check that

$$\forall t' \in \Omega(\varphi, t) \quad (s \in \mathbf{S}_{\mathcal{F}}(t')). \tag{6}$$

If $\varphi = 0$, inequality $\rho \ge \varphi$ is evident. Therefore, hereafter it is supposed that $\varphi > 0$.

Following the definitions of $\varphi(k), j_k^*$, and representation (5) of $\Theta(\cdot)$, it is not difficult to see that for each $k \in \widetilde{W}^s$ we have the following:

$$\forall i \in \mathcal{I}_{j_k^*} \quad (\Delta_{kl_i} \le \varphi(k)), \tag{7}$$

$$\forall i \in \mathcal{I}_{|\widetilde{B}_{k>}|} \setminus \mathcal{I}_{j_k^*} \quad (\varphi(k) < \Delta_{kl_i}). \tag{8}$$

Whence, in view of Property 1, Lemma 1, and (4), for any $t' \in \Omega(\varphi(k), t)$, $k \in \widetilde{W}^s$, we derive

$$\begin{aligned} t'_w(V_k) &= \sum_{l \in B_k} t'_b(V_{kl}) = \sum_{l \in B_k \setminus \widetilde{B}_k} t_b(V_{kl}) + \sum_{l \in \widetilde{B}_{k0}} t'_b(V_{kl}) + \sum_{l \in \widetilde{B}_{k>}} t'_b(V_{kl}) \\ &\le \sum_{l \in B_k \setminus \widetilde{B}_k} t_b(V_{kl}) + \sum_{l \in \widetilde{B}_{k0}} (t_b(V_{kl}) + \varphi(k)) + \sum_{i \in \mathcal{I}_{j_k^*}} (t_b(V_{kl_i}) + \varphi(k) - \Delta_{kl_i}) \\ &\quad + \sum_{i \in \mathcal{I}_{|\widetilde{B}_{k>}|} \setminus \mathcal{I}_{j_k^*}} t_b(V_{kl_i}) = t_w(V_k) + \varphi(k)(|\widetilde{B}_{k0}| + j_k^*) - \sum_{i \in \mathcal{I}_{j_k^*}} \Delta_{kl_i} = T_0. \end{aligned}$$

Hence, the following formula is proven:

$$\forall k \in \widetilde{W}^s\, \forall t' \in \Omega(\varphi(k), t) \quad (t'_w(V_k) \le T_0).$$

Therefore, taking into account the following obvious inequalities:

$$\forall \varepsilon > 0\, \forall k \in W^s \setminus \widetilde{W}^s\, \forall t' \in \Omega(\varepsilon, t) \quad (t'_w(V_k) = t_w(V_k) \le T_0),$$

we conclude that

$$\forall k \in W^s\, \forall t' \in \Omega(\varphi, t) \quad (t'_w(V_k) \le T_0)$$

and this implies (6).

Now let us show that $\rho \le \varphi$. The proof of the latter inequality is equivalent to the proof of the following formula:

$$\forall \varepsilon > \varphi\, \exists t^* \in \Omega(\varepsilon, t) \quad (s \notin \mathbf{S}_{\mathcal{F}}(t^*)). \tag{9}$$

To prove formula (9), the definition of φ is used. By definition of φ there is $k^* \in \widetilde{W}^s$ such that $\varphi = \varphi(k^*)$. Then, assuming $\varepsilon > \varphi$, $t^* \in \Omega(\varepsilon, t)$, where

$$t^*_j = \begin{cases} t_j + \delta, & \text{if } j \in \widetilde{V}_{k^*}, \\ t_j & \text{otherwise,} \end{cases}$$

$$\varphi < \delta < \min\{\varepsilon, \xi(k^*)\},$$

$$\xi(k^*) = \min_{i \in \mathcal{I}_{|\widetilde{B}_{k^*>}|} \setminus \mathcal{I}_{j^*_{k^*}}} \Delta_{k^* l_i},$$

and taking into account (4), (7) and (8), we obtain

$$\begin{aligned} t^*_w(V_{k^*}) &= \sum_{l \in B_{k^*}} t^*_b(V_{k^*l}) = \sum_{l \in B_{k^*} \setminus \widetilde{B}_{k^*}} t_b(V_{k^*l}) + \sum_{l \in \widetilde{B}_{k^*0}} t^*_b(V_{k^*l}) + \sum_{l \in \widetilde{B}_{k^*>}} t^*_b(V_{k^*l}) \\ &= \sum_{l \in B_{k^*} \setminus \widetilde{B}_{k^*}} t_b(V_{k^*l}) + \sum_{l \in \widetilde{B}_{k^*0}} (t_b(V_{k^*l}) + \delta) + \sum_{i \in \mathcal{I}_{j^*_{k^*}}} (t_b(V_{k^*l_i}) + \delta - \Delta_{k^*l_i}) \\ &\quad + \sum_{i \in \mathcal{I}_{|\widetilde{B}_{k^*>}|} \setminus \mathcal{I}_{j^*_{k^*}}} t_b(V_{k^*l_i}) > t_w(V_{k^*}) + \varphi(k^*)(|\widetilde{B}_{k^*0}| + j^*_{k^*}) - \sum_{i \in \mathcal{I}_{j^*_{k^*}}} \Delta_{k^*l_i} = T_0. \end{aligned}$$

It follows that $s \notin \mathbf{S}_{\mathcal{F}}(t^*)$, and therefore (9) holds. $\square$

Theorem 1 implies

Corollary 1. *Solution $s \in \mathbf{S}_{\mathcal{F}}(t)$ is not $\mathcal{F}$-stable iff the following holds:*

$$\exists k \in \widetilde{W}^s \quad (t_w(V_k) = T_0 \text{ and } \widetilde{B}_{k0} \neq \emptyset). \tag{10}$$

Proof (*Sufficiency*)**.** Let $k \in \widetilde{W}^s$ be a workstation that satisfies condition (10). Then, following the definition of j^*_k, we obtain $j^*_k = 0$ and, as a consequence, $\varphi(k) = 0$. The latter, according to (3), implies that $\rho_{\mathcal{F}}(s, t) = 0$ and therefore s is not $\mathcal{F}$-stable.

Necessity. Assume the contrary. Let s be not $\mathcal{F}$-stable, while

$$\forall k \in \widetilde{W}^s \quad (t_w(V_k) < T_0 \vee \widetilde{B}_{k0} = \emptyset) \tag{11}$$

holds. Let us consider two possible cases for an arbitrary $k \in \widetilde{W}^s$.

Case 1. $t_w(V_k) = T_0$. Then, following (11), $\widetilde{B}_{k0} = \emptyset$ holds. This yields $j^*_k > 0$ due to (4). Whence $\varphi(k) > 0$.

Case 2. $t_w(V_k) < T_0$. In this case, following (4), we conclude once again that $\varphi(k) > 0$.

Thus, summarizing these two cases, we have $\rho_{\mathcal{F}}(s, t) > 0$ due to (3). This implies that s is $\mathcal{F}$-stable and this is contradictory with the initial assumption. $\square$

Corollary 2. *The $\mathcal{F}$-stability radius of solution $s \in \mathbf{S}_{\mathcal{F}}(t)$ can be calculated in $\mathcal{O}(n + \sum_{k \in \widetilde{W}^s} |\widetilde{B}_{k>}| \log |\widetilde{B}_{k>}|)$ time.*

Proof. An algorithm to find $\rho_{\mathcal{F}}(s, t)$ is based on a sequential analysis of the workstations of solution s. Thus, for each block l of current workstation $k \in W^s$, the following characteristics are calculated: block time $t_b(V_{kl})$ and Δ_{kl}, if $l \in \widetilde{B}_k$ (this takes $\mathcal{O}(|V_{kl}|)$ computing time). In the case, where all blocks of current workstation k are examined and no uncertain task is found, the next workstation is analyzed, otherwise $\varphi(k)$ is calculated. To calculate $\varphi(k)$, values Δ_{kl}, $l \in \widetilde{B}_{k>}$, are ordered in the non-decreasing order (this takes $\mathcal{O}(|\widetilde{B}_{k>}| \log |\widetilde{B}_{k>}|)$ computing time) and index j^*_k is found at most in $\mathcal{O}(|\widetilde{B}_{k>}|)$ time due to representation (5) of $\Theta(\cdot)$. This sequence of computations is reasonable only in the case where $\widetilde{B}_{k>} \neq \emptyset$, $\varphi(k) = \frac{T_0 - t_w(V_k)}{|\widetilde{B}_{k0}|}$ otherwise. This continues until either all workstations are analyzed or workstation $k \in \widetilde{W}^s$ such that $\varphi(k) = 0$ is found.

In the latter case $\rho_{\mathcal{F}}(s,t) = 0$, otherwise $\rho_{\mathcal{F}}(s,t) = \min_{k\in\widetilde{W}^s}\varphi(k)$. Thus, in the worst case, where no $\varphi(k)$, $k \in \widetilde{W}^s$, equal to 0, and no $\widetilde{B}_{k>}$, $k \in \widetilde{W}^s$, is an empty set, this algorithm takes $\mathcal{O}(\sum_{k\in W^s}\sum_{l\in B_k}|V_{kl}| + \sum_{k\in\widetilde{W}^s}(|\widetilde{B}_{k>}|\log|\widetilde{B}_{k>}| + |\widetilde{B}_{k>}|)) = \mathcal{O}(n + \sum_{k\in\widetilde{W}^s}|\widetilde{B}_{k>}|\log|\widetilde{B}_{k>}|)$ time. $\square$

Taking into account Corollaries 1 and 2, and representation (5), the general scheme of the $\mathcal{F}$-stability radius calculation is represented by Algorithm 1, where binary variable *flag* is used to control if a new iteration is needed.

It is easy to see that if the following conditions hold:

- only one task per block is authorized,
- number of blocks per workstation is not limited,
- there is no exclusion constraints,
- $C_2 = 0$,

the considered line balancing problem is reduced to the well-known SALBP-1. In this case, $\widetilde{B}_{k>}$ is an empty set, and therefore, $j_k^* = 0$ for each $k \in \widetilde{W}^s$. Moreover, $\widetilde{B}_{k0} = \widetilde{V}_k$ holds. Thus, Theorem 1 and Corollary 1 imply the results obtained in [28], i.e. Corollaries 3 and 4:

Corollary 3. *The $\mathcal{F}$-stability radius of solution $s \in \mathbf{S}_{\mathcal{F}}(t)$ for SALBP-1 is calculated as follows:*

$$\rho_{\mathcal{F}}(s,t) = \min_{k\in\widetilde{W}^s}\frac{T_0 - t_w(V_k)}{|\widetilde{V}_k|}.$$

Corollary 4. *Solution $s \in \mathbf{S}_{\mathcal{F}}(t)$ of SALBP-1 is not $\mathcal{F}$-stable iff there exists $k \in \widetilde{W}$ such that $T_0 = t_w(V_k)$.*

Algorithm 1: $\mathcal{F}$-stability radius calculation

flag $\leftarrow$ true /* While s is $\mathcal{F}$-stable */
foreach $k \in \widetilde{W}^s$ **do**
  if $t_w(V_k) = T_0$ **and** $\widetilde{B}_{k0} \neq \emptyset$ **then**
    $\rho_{\mathcal{F}}(s,t) \leftarrow 0$, *flag* $\leftarrow$ false /* s is not $\mathcal{F}$-stable */
    break /* Forcibly terminate loop **foreach** */
  else if $\widetilde{B}_{k>} = \emptyset$ **then**
    $\varphi(k) \leftarrow \frac{T_0 - t_w(V_k)}{|\widetilde{B}_{k0}|}$
  else
    Sort Δ_{kl}, $l \in \widetilde{B}_{k>}$: $0 = \Delta_{kl_0} < \Delta_{kl_1} \le \ldots \le \Delta_{kl_{|\widetilde{B}_{k>}|}}$
    $j \leftarrow 0$, $\varphi(k) \leftarrow 0$, $\Theta \leftarrow 0$
    while $j < |\widetilde{B}_{k>}|$ **do**
      if $\Theta + (|\widetilde{B}_{k0}| + j)(\Delta_{kl_{j+1}} - \Delta_{kl_j}) \le T_0 - t_w(V_k)$ **then**
        $\Theta \leftarrow \Theta + (|\widetilde{B}_{k0}| + j)(\Delta_{kl_{j+1}} - \Delta_{kl_j})$, $\varphi(k) \leftarrow \Delta_{kl_{j+1}}$, $j \leftarrow j+1$
      else
        break /* Forcibly terminate loop **while** */
    $\varphi(k) \leftarrow \varphi(k) + \frac{T_0 - t_w(V_k) - \Theta}{|\widetilde{B}_{k0}| + j}$
if *flag* $=$ true **then**
  $\rho_{\mathcal{F}}(s,t) \leftarrow \min\{\varphi(k) : k \in \widetilde{W}^s\}$

An example is used to illustrate the $\widetilde{\mathcal{F}}$-stability radius calculation.

Let $V = \{1, 2, \ldots, 8\}$, $\widetilde{V} = \{1, 3, 5\}$, $t = (0.5, 1, 0.75, 1, 1.5, 3.5, 1, 1)$, $T_0 = 5$, $b_0 = 2$, $E^w = \emptyset$, $E^b = \emptyset$. The precedence constraints are represented by the acyclic directed graph shown in Fig. 2, where the uncertain tasks are dotted.

Let s be a feasible solution with two workstations and two blocks per workstation such that $V_{11}^s = \{1, 2\}$, $V_{12}^s = \{3, 4\}$, $V_{21}^s = \{5, 6\}$, $V_{22}^s = \{7, 8\}$ (see Fig. 3).

It is easy to see that $\widetilde{W}^s = \{1, 2\}$, $\widetilde{B}_1 = \widetilde{B}_{1>} = \{1, 2\}$, $\widetilde{B}_{10} = \emptyset$, $\widetilde{B}_2 = \widetilde{B}_{2>} = \{1\}$, $\widetilde{B}_{20} = \emptyset$ and $t_b(V_{11}^s) = 1$, $t_b(V_{12}^s) = 1$, $t_b(V_{21}^s) = 3.5$, $t_b(V_{22}^s) = 1$. As a consequence, $\Delta_{11} = 0.5$, $\Delta_{12} = 0.25$, $\Delta_{21} = 2$ and $t_w(V_1^s) = 2$, $t_w(V_2^s) = 4.5$. Therefore, using Theorem 1, we obtain

$$\varphi(1) = 0.5 + \frac{5 - 2 - 0.25}{2} = 1.875, \qquad \varphi(2) = 2 + \frac{5 - 4.5 - 0}{1} = 2.5.$$

Thus, $\rho_{\mathcal{F}}(s,t) = 1.875$.

In the next section, the $\widehat{\mathcal{F}}$-stability radius for quasi-feasible solutions is evaluated.

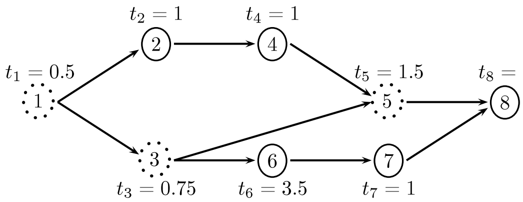

Fig. 2. Precedence constraints.

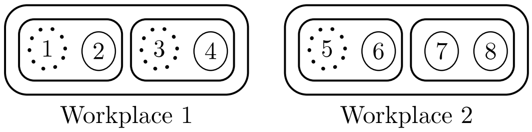

Fig. 3. A feasible solution.

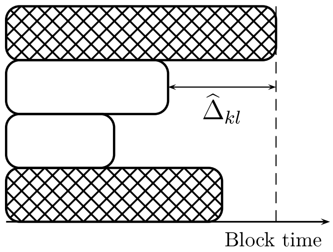

Fig. 4. Block l of workstation k of solution s.

4. Stability analysis for quasi-feasible solutions

Recall that for any solution $s \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)$ there exists workstation $k \in W^s$ such that $t_w(V_k) > T_0$ holds. Despite of this, a quasi-feasible solution may become a feasible one. This may occur if for a new perturbed vector $t^* \in \mathbb{R}^n_+$, $t^*_w(V_k) \le T_0$ holds for any $k \in W^s$. However, if for a quasi-feasible solution, the cycle time constraint is violated for a workstation without uncertain tasks, this solution always remains quasi-feasible. In the sequel, the set of workstations of solution s having working time exceeding T_0 is denoted by $\widehat{W}^s$.

The following notations will be used for each $s \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)$ and $k \in \widehat{W}^s$:

- $\widehat{\Delta}_{kl} = t_b(\widetilde{V}_{kl}) - t_b(V_{kl} \setminus \widetilde{V}_{kl})$, $l \in \widetilde{B}_k$, is the difference between the maximal processing times calculated for uncertain and constant tasks assigned to block l,
- $\widehat{B}_k = \{l \in \widetilde{B}_k : \widehat{\Delta}_{kl} > 0\}$ is the set of blocks where $\widehat{\Delta}_{kl} > 0$, i.e. the difference between the maximal processing times calculated for uncertain and constant tasks assigned to block l is positive.

An illustrative example is given in Fig. 4, where a block with four tasks is shown. Each task is represented by a horizontal column which length corresponds to its processing time. Uncertain tasks are cross-hatched.

Property 2. *If for workstation $k \in \widehat{W}^s$ of solution $s \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)$ inequality $\widetilde{B}_k \setminus \widehat{B}_k \neq \emptyset$ holds, then*

$$\forall \varepsilon > 0\, \forall t' \in \Omega(\varepsilon, t)\, \forall l \in \widetilde{B}_k \setminus \widehat{B}_k \qquad (t'_b(V_{kl}) \ge t_b(V_{kl})).$$

Lemma 2. *If for $s \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)$ there exists $k \in \widehat{W}^s$ so that $\widehat{B}_k \neq \emptyset$, then*

$$\forall l \in \widehat{B}_k\, \forall \varepsilon > 0\, \forall t' \in \Omega(\varepsilon, t) \qquad (t'_b(V_{kl}) \ge t_b(V_{kl}) - \widehat{\Delta}_{kl}), \tag{12}$$

$$\forall l \in \widehat{B}_k\, \forall \varepsilon < \widehat{\Delta}_{kl}\, \forall t' \in \Omega(\varepsilon, t) \qquad (t'_b(V_{kl}) > t_b(V_{kl}) - \varepsilon). \tag{13}$$

Proof. By definition

$$t'_b(V_{kl}) = \max\{t'_b(\widetilde{V}_{kl}), t_b(V_{kl} \setminus \widetilde{V}_{kl})\}.$$

Therefore, taking into account obvious equalities

$$t_b(\widetilde{V}_{kl}) = t_b(V_{kl}), \quad l \in \widehat{B}_k,$$

and Property 1, we obtain

$$t'_b(V_{kl}) \geq t_b(V_{kl} \setminus \widetilde{V}_{kl}) = t_b(\widetilde{V}_{kl}) - \widehat{\Delta}_{kl} = t_b(V_{kl}) - \widehat{\Delta}_{kl}, \quad t' \in \Omega(\varepsilon, t),$$
$$t'_b(\widetilde{V}_{kl}) > t_b(\widetilde{V}_{kl}) - \varepsilon > t_b(\widetilde{V}_{kl}) - \widehat{\Delta}_{kl} = t_b(V_{kl} \setminus \widetilde{V}_{kl}), \quad t' \in \Omega(\varepsilon, t),$$

and this implies the validity of (12) and (13). □

Theorem 2. *The $\widehat{\mathcal{F}}$-stability radius of $s \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)$ is calculated as follows:*

$$\rho_{\widehat{\mathcal{F}}}(s, t) = \max_{k \in \widehat{W}^s} \widehat{\varphi}(k), \tag{14}$$

where

$$\widehat{\varphi}(k) = \widehat{\Delta}_{kl_{j_k^*}} + \frac{t_w(V_k) - T_0 - \widehat{\Theta}(j_k^*)}{|\widehat{B}_k| - j_k^*},$$
$$j_k^* = \operatorname{argmax}\{\widehat{\Theta}(j) : \widehat{\Theta}(j) \leq t_w(V_k) - T_0, \quad j \in \{0\} \cup \mathcal{I}_{|\widehat{B}_k|}\},$$
$$\widehat{\Theta}(j) = (|\widehat{B}_k| - j)\widehat{\Delta}_{kl_j} + \sum_{i \in \mathcal{I}_j} \widehat{\Delta}_{kl_i},$$
$$0 = \widehat{\Delta}_{kl_0} \leq \cdots \leq \widehat{\Delta}_{kl_{|\widehat{B}_k|}}, \qquad l_i \in \widehat{B}_k, \quad i \in \mathcal{I}_{|\widehat{B}_k|}.$$

Proof. As in Theorem 1, supplementary notations are introduced: $\widehat{\rho}$ and $\widehat{\varphi}$ are the left-hand and right-hand sides of (14), respectively. It is easy to see that $\widehat{\varphi}$ can be equal to $+\infty$. This may occur if $|\widehat{B}_k| = j_k^*$ for some index $k \in \widehat{W}^s$ and means that solution s will never become feasible. Otherwise $\widehat{\varphi}$ is a positive finite number due to inclusion $s \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)$ and the definition of j_k^*. Therefore, hereafter we suppose that

$$|\widehat{B}_k| - j_k^* > 0, \quad k \in \widehat{W}^s. \tag{15}$$

Remark also that non-decreasing function $\widehat{\Theta}(\cdot)$ has a useful recursive representation:

$$\widehat{\Theta}(j+1) = \widehat{\Theta}(j) + (|\widehat{B}_k| - j)(\widehat{\Delta}_{kl_{j+1}} - \widehat{\Delta}_{kl_j}), \qquad \Theta(0) = 0. \tag{16}$$

To prove formula (14), we consequently show that inequalities $\widehat{\rho} \geq \widehat{\varphi}$ and $\widehat{\rho} \leq \widehat{\varphi}$ hold. First, let us prove that $\widehat{\rho} \geq \widehat{\varphi}$. To do this, it is sufficient to check that

$$\forall t' \in \Omega(\widehat{\varphi}, t) \quad (s \in \mathbf{S}_{\widehat{\mathcal{F}}}(t')). \tag{17}$$

If $\widehat{\varphi} = 0$, inequality $\widehat{\rho} \geq \widehat{\varphi}$ is evident. Therefore, hereafter it is supposed that $\widehat{\varphi} > 0$.

Following the definitions of $\widehat{\varphi}(k), j_k^*$, and representation (16) of $\widehat{\Theta}(\cdot)$, it is not difficult to see that for any $k \in \widehat{W}^s$ we have

$$\forall i \in \mathcal{I}_{j_k^*} \quad (\widehat{\Delta}_{kl_i} \leq \widehat{\varphi}(k)), \tag{18}$$

$$\forall i \in \mathcal{I}_{|\widehat{B}_k|} \setminus \mathcal{I}_{j_k^*} \quad (\widehat{\varphi}(k) < \widehat{\Delta}_{kl_i}), \tag{19}$$

and there exists $k^* \in \widehat{W}^s$ such that $\widehat{\varphi}(k^*) = \widehat{\varphi}$.

Whence, in view of Property 2, Lemma 2, and (15), for any $t' \in \Omega(\widehat{\varphi}, t)$ we derive

$$\begin{aligned} t'_w(V_{k^*}) &= \sum_{l \in B_{k^*}} t'_b(V_{k^*l}) = \sum_{l \in B_{k^*} \setminus \widetilde{B}_{k^*}} t_b(V_{k^*l}) + \sum_{l \in \widetilde{B}_{k^*} \setminus \widehat{B}_{k^*}} t'_b(V_{k^*l}) + \sum_{l \in \widehat{B}_{k^*}} t'_b(V_{k^*l}) \\ &> \sum_{l \in B_{k^*} \setminus \widetilde{B}_{k^*}} t_b(V_{k^*l}) + \sum_{l \in \widetilde{B}_{k^*} \setminus \widehat{B}_{k^*}} t_b(V_{k^*l}) + \sum_{i \in \mathcal{I}_{j_{k^*}^*}} (t_b(V_{k^*l_i}) - \widehat{\Delta}_{k^*l_i}) \\ &\quad + \sum_{i \in \mathcal{I}_{|\widehat{B}_{k^*}|} \setminus \mathcal{I}_{j_{k^*}^*}} (t_b(V_{k^*l_i}) - \widehat{\varphi}(k^*)) = t_w(V_{k^*}) - \sum_{i \in \mathcal{I}_{j_{k^*}^*}} \widehat{\Delta}_{k^*l_i} - \widehat{\varphi}(k^*)(|\widehat{B}_{k^*}| - j_{k^*}^*) = T_0. \end{aligned}$$

Hence, the following formula is proven

$$\exists k^* \in \widehat{W}^s \, \forall t' \in \Omega(\widehat{\varphi}, t) \quad (t'_w(V_{k^*}) > T_0),$$

this implies (17).

Now, let us show that $\widehat{\rho} \leq \widehat{\varphi}$. The proof of the latter inequality is equivalent to the proof of the following formula:

$$\forall \varepsilon > \widehat{\varphi} \, \exists t^* \in \Omega(\varepsilon, t) \quad (s \in \mathbf{S}_{\mathcal{F}}(t^*)). \tag{20}$$

To prove formula (20), the definition of $\widehat{\varphi}$ is used. By definition of $\widehat{\varphi}$ for any $k \in \widehat{W}^s$ we have $\widehat{\varphi} \geq \widehat{\varphi}(k)$. Then, assuming $\varepsilon > \widehat{\varphi}$, $t^* \in \Omega(\varepsilon, t)$, where

$$t_j^* = \begin{cases} t_j - \delta, & \text{if } j \in \widetilde{V}_{kl}, l \in \widehat{B}_k, k \in \widehat{W}^s, \\ t_j & \text{otherwise,} \end{cases}$$
$$\widehat{\varphi} < \delta < \min\{\varepsilon, \xi\},$$
$$\xi = \min_{k \in \widehat{W}^s} \min_{i \in \mathcal{I}_{|\widehat{B}_k|} \setminus \mathcal{I}_{j_k^*}} \widehat{\Delta}_{kl_i},$$

and taking into account (15), (18) and (19), we obtain

$$\begin{aligned} t_w^*(V_k) &= \sum_{l \in B_k} t_b^*(V_{kl}) = \sum_{l \in B_k \setminus \widetilde{B}_k} t_b(V_{kl}) + \sum_{l \in \widetilde{B}_k \setminus \widehat{B}_k} t_b(V_{kl}) + \sum_{l \in \widehat{B}_k} t_b^*(V_{kl}) \\ &= \sum_{l \in B_k \setminus \widetilde{B}_k} t_b(V_{kl}) + \sum_{l \in \widetilde{B}_k \setminus \widehat{B}_k} t_b(V_{kl}) + \sum_{i \in \mathcal{I}_{j_k^*}} (t_b(V_{kl_i}) - \widehat{\Delta}_{kl_i}) \\ &\quad + \sum_{i \in \mathcal{I}_{|\widehat{B}_k|} \setminus \mathcal{I}_{j_k^*}} (t_b(V_{kl_i}) - \delta) > t_w(V_k) - \widehat{\varphi}(k)(|\widehat{B}_k| - j_k^*) - \sum_{i \in \mathcal{I}_{j_k^*}} \widehat{\Delta}_{kl_i} = T_0, \ k \in \widehat{W}^s. \end{aligned}$$

Therefore, taking into account the following obvious inequalities $t_w^*(V_k) \leq T_0$, $k \in W^s \setminus \widehat{W}^s$, we conclude that

$$\forall \varepsilon > \widehat{\varphi} \, \exists t^* \in \Omega(\varepsilon, t) \, \forall k \in W^s \quad (t_w^*(V_k) \leq T_0).$$

In other words, formula (20) holds. □

Theorem 2 implies Corollaries 5 and 6.

Corollary 5. *Any quasi-feasible solution is $\widehat{\mathcal{F}}$-stable.*

Proof. Since for any $s \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)$, inequalities $t_w(V_k) > T_0$, $k \in \widehat{W}^s$, hold, then according to the definition of j_k^*, equality $\widehat{\Theta}(j_k^*) = t_w^*(V_k) - T_0$ is possible only for $j_k^* > 0$. Therefore, following the definition of $\widehat{\varphi}(k)$, we have $\widehat{\varphi}(k) > 0$, $k \in \widehat{W}^s$, this implies $\rho_{\widehat{\mathcal{F}}}(s, t) > 0$ due to (14). □

Corollary 6. *$\widehat{\mathcal{F}}$-stability radius of solution $s \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)$ can be calculated in $\mathcal{O}(n + \sum_{k \in \widehat{W}^s} |\widehat{B}_k| \log |\widehat{B}_k|)$ time.*

Proof. An approach similar to that proposed for Corollary 2 is used. An algorithm to find $\rho_{\widehat{\mathcal{F}}}(s, t)$ is based on a sequential analysis of the workstations of solution s. For each block l of current workstation $k \in W^s$, block time $t_b(V_{kl})$ is calculated as well as $\widehat{\Delta}_{kl}$ if block l has uncertain tasks (this takes $\mathcal{O}(|V_{kl}|)$ computing time). If the cycle time constraint is respected for workstation k, then the next workstation is analyzed, otherwise $\widehat{\varphi}(k)$ is calculated. To do it, values $\widehat{\Delta}_{kl}$, $l \in \widehat{B}_k$, are ordered in non-decreasing order (this takes $\mathcal{O}(|\widehat{B}_k| \log |\widehat{B}_k|)$ computing time) and index j_k^* is found at most in $\mathcal{O}(|\widehat{B}_k|)$ time due to representation (16) of $\widehat{\Theta}(\cdot)$. This algorithm is reasonable only in the case where $\widehat{B}_k \neq \emptyset$, $\widehat{\varphi}(k) = +\infty$ otherwise. This continues until either all workstations are analyzed or workstation $k \in \widehat{W}^s$ such that $\widehat{\varphi}(k) = +\infty$ is found. In the latter case $\rho_{\widehat{\mathcal{F}}}(s, t) = +\infty$, otherwise $\rho_{\widehat{\mathcal{F}}}(s, t) = \max_{k \in \widehat{W}^s} \widehat{\varphi}(k)$. Thus, in the worst case, where no $\widehat{\varphi}(k)$, $k \in \widehat{W}^s$, equal to $+\infty$ and no $\widehat{B}_k$, $k \in \widehat{W}^s$, is an empty set, this algorithm takes $\mathcal{O}(\sum_{k \in W^s} \sum_{l \in B_k} |V_{kl}| + \sum_{k \in \widehat{W}^s} (|\widehat{B}_k| \log |\widehat{B}_k| + |\widehat{B}_k|)) = \mathcal{O}(n + \sum_{k \in \widehat{W}^s} |\widehat{B}_k| \log |\widehat{B}_k|)$ time. □

Taking into account Corollary 6 and representation (16), the general scheme of the $\widehat{\mathcal{F}}$-stability radius calculation is given by Algorithm 2, where binary variable *flag* is used to control if a new iteration is needed.

As in Section 3, it can be remarked that for SALBP-1 equality $\widehat{B}_k = \widetilde{V}_k$ holds for each $k \in \widehat{W}^s$. Thus, Theorem 2 and Corollary 5 imply the results obtained in [28], i.e. Corollaries 7 and 8:

Corollary 7. *The $\widehat{\mathcal{F}}$-stability radius of solution $s \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)$ for SALBP-1 is calculated as follows:*

$$\rho_{\widehat{\mathcal{F}}}(s, t) = \max_{k \in \widehat{W}^s} \max_{j \in \{0\} \cup \mathcal{I}_{|\widetilde{V}_k|}} \frac{t_w(V_k) - T_0 - \sum_{i \in \mathcal{I}_j} t_{l_i}}{|\widetilde{V}_k| - j},$$

where

$$0 = t_{l_0} \leq \cdots \leq t_{l_{|\widetilde{V}_k|}}, \qquad l_i \in \widetilde{V}_k, \quad i \in \mathcal{I}_{|\widetilde{V}_k|}.$$

Corollary 8. *Any quasi-feasible solution of SALBP-1 is $\widehat{\mathcal{F}}$-stable.*

Algorithm 2: $\widehat{\mathcal{F}}$-stability radius calculation

flag $\leftarrow$ true /* While s can become feasible */
foreach $k \in \widehat{W}^s$ **do**
  if $\widetilde{B}_k = \emptyset$ **or** $\widehat{B}_k = \emptyset$ **then**
    $\rho_{\widehat{\mathcal{F}}}(s,t) \leftarrow +\infty$
    flag $\leftarrow$ false /* s cannot become feasible */
    break /* Forcibly terminate loop **foreach** */
  else
    Sort $\widehat{\Delta}_{kl}, l \in \widehat{B}_k$: $0 = \widehat{\Delta}_{kl_0} < \widehat{\Delta}_{kl_1} \leq \ldots \leq \widehat{\Delta}_{kl_{|\widehat{B}_k|}}$
    $j \leftarrow 0, \widehat{\varphi}(k) \leftarrow 0, \widehat{\Theta} \leftarrow 0$
    while $j < |\widehat{B}_k|$ **do**
      if $\widehat{\Theta} + (|\widehat{B}_k| - j)(\widehat{\Delta}_{kl_{j+1}} - \widehat{\Delta}_{kl_j}) \leq t_w(V_k) - T_0$ **then**
        $\widehat{\Theta} = \widehat{\Theta} + (|\widehat{B}_k| - j)(\widehat{\Delta}_{kl_{j+1}} - \widehat{\Delta}_{kl_j}), \widehat{\varphi}(k) \leftarrow \widehat{\Delta}_{kl_{j+1}}, j \leftarrow j+1$
      else
        break /* Forcibly terminate loop **while** */
    if $j < |\widehat{B}_k|$ **then**
      $\widehat{\varphi}(k) \leftarrow \widehat{\varphi}(k) + \frac{t_w(V_k) - T_0 - \widehat{\Theta}}{|\widehat{B}_k| - j}$
    else
      $\rho_{\widehat{\mathcal{F}}}(s,t) \leftarrow +\infty$
      flag $\leftarrow$ false /* s cannot become feasible */
      break /* Forcibly terminate loop **foreach** */
if *flag* $=$ true **then**
  $\rho_{\widehat{\mathcal{F}}}(s,t) \leftarrow \max\{\widehat{\varphi}(k) : k \in \widehat{W}^s\}$

5. Stability analysis for optimal solutions

There are two principal cases where an optimal solution s can lose its optimality for a new perturbed vector t^*:

- s loses its feasibility (this case was studied in Section 3);
- s remains feasible, but there is a quasi-feasible solution s^0 with $\mathcal{Z}(s^0, t) < \mathcal{Z}(s, t)$ such that $s \in \mathbf{S}_{\mathcal{F}}(t^*)$ (this case was studied in Section 4).

Therefore, Theorems 1 and 2 imply

Theorem 3. *$\mathcal{O}$-stability radius $\rho_{\mathcal{O}}(s, t)$ of solution $s \in \mathbf{S}_{\mathcal{O}}(t)$ is calculated as follows:*

$$\rho_{\mathcal{O}}(s,t) = \min\left\{\rho_{\mathcal{F}}(s,t), \min_{s' \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)} \{\rho_{\widehat{\mathcal{F}}}(s',t) : \mathcal{Z}(s',t) < \mathcal{Z}(s,t)\}\right\}.$$

Nevertheless, the problem of finding $\mathcal{O}$-stability radius $\rho_{\mathcal{O}}(s, t)$ remains difficult, since it is necessary to know the whole set of quasi-feasible solutions with the value of the objective function smaller than this for solution s.

Corollaries 1 and 5 imply:

Corollary 9. *Solution $s \in \mathbf{S}_{\mathcal{O}}(t)$ is not $\mathcal{O}$-stable iff the following holds:*

$$\exists k \in \widetilde{W}^s \quad (t_w(V_k) = T_0 \text{ and } \widetilde{B}_{k0} \neq \emptyset). \tag{21}$$

In the case of SALBP-1, Theorem 3 and Corollary 9 imply the results obtained in [28], i.e. Corollaries 10 and 11:

Corollary 10. *$\mathcal{O}$-stability radius $\rho_{\mathcal{O}}(s, t)$ of solution $s \in \mathbf{S}_{\mathcal{O}}(t)$ for SALBP-1 is calculated as follows:*

$$\rho_{\mathcal{O}}(s,t) = \min\left\{\rho_{\mathcal{F}}(s,t), \min_{s' \in \mathbf{S}_{\widehat{\mathcal{F}}}(t)} \{\rho_{\widehat{\mathcal{F}}}(s',t) : |W^{s'}| < |W^s|\}\right\}.$$

Corollary 11. *Solution $s \in \mathbf{S}_{\mathcal{O}}(t)$ of SALBP-1 is not $\mathcal{O}$-stable iff there exists $k \in \widetilde{W}^s$ such that $T_0 = t_w(V_k)$.*

The proposed stability radius can be used as a measure of the solution robustness. Therefore, the initial mono-objective problem can be reformulated as a multi-objective one, where solution cost is to minimize and at the same time the solution robustness is to maximize. This problem is addressed in the next section.

6. Multi-objective optimization

6.1. General approach

Taking into account the stability measure suggested in this paper, the initial optimization problem becomes multi-objective aiming to optimize the line cost and its robustness simultaneously. These two objectives are contradictory, therefore, to evaluate the solutions obtained, the concept of Pareto-optimality [11] is used. A solution is called efficient or non-dominated if no other solution having better values of both criteria exists. To formally present the set of non-dominated solutions ($\mathcal{NDS}$), the binary relation between any two feasible solutions s and s' reflecting the Pareto dominant rule is introduced as follows:

$$s \succ s' \Longleftrightarrow \mathcal{Z}(s,t) \le \mathcal{Z}(s',t) \quad \text{and} \quad \rho_{\mathcal{F}}(s,t) \ge \rho_{\mathcal{F}}(s',t),$$

where strict inequality holds at least once. In the case where $s \succ s'$ we say that s dominates s' or s' is dominated by s.

Thus,

$$\mathcal{NDS} = \{s \in \mathbf{S}_{\mathcal{F}}(t) : \nexists s' \in \mathbf{S}_{\mathcal{F}}(t)\ (s' \succ s)\}.$$

To obtain set $\mathcal{NDS}$ is a non-polynomially solvable problem. In this paper, a multistage approximate algorithm is used to construct an approximation $\mathcal{ANDS}$ of $\mathcal{NDS}$. At each stage of this algorithm, a multi-start heuristic procedure $\mathcal{H}(\rho)$ that constructs a feasible solution whose $\mathcal{F}$-stability radius is greater than the current value of ρ is applied over $T_{\max}$ time-period. Among the solutions constructed at the current stage, those with the minimal cost are considered and the solution with the greatest $\mathcal{F}$-stability radius is chosen among them. The value of its $\mathcal{F}$-stability radius becomes the current value of ρ for the next stage. This continues until $\rho < T_0 - \tilde{t}_{\max}$ holds, where $\tilde{t}_{\max} = \max\{t_j : j \in \widetilde{V}\}$. At the beginning of the algorithm, $\rho = -1$.

Let $\mathcal{S}$ be the set of feasible solutions chosen at each stage of the algorithm. Then, the used approximation $\mathcal{ANDS}$ of $\mathcal{NDS}$ can be expressed as follows:

$$\mathcal{ANDS} = \{s \in \mathcal{S} : \nexists s' \in \mathcal{S}\ (s' \succ s)\}.$$

6.2. Heuristic $\mathcal{H}(\rho)$

Given a current value of ρ, this heuristic constructs a feasible solution by assigning as many tasks as possible to the current block of the current workstation. At the beginning, a feasible solution contains only one workstation with one empty block. The heuristic assigns tasks to this block until no task can be added because of the existing constraints. Then, a new empty block (or a new workstation with one empty block) is opened and becomes current. This continues until all tasks are assigned and a feasible solution is obtained.

Let k and l be respectively the current workstation and the current block of solution s. To choose an task to be assigned to V_{kl}, the so-called Candidate List $\mathcal{CL}(k,l)$ of tasks is generated. It contains all tasks that can be assigned to block l of workstation k. This list is built in the following way: the set of unassigned tasks is analyzed and task j is added to $\mathcal{CL}(k,l)$ if all following conditions are satisfied:

- all predecessors of j have been already assigned,
- j is not linked by exclusion constraints with other tasks already assigned to the current block and to the current workstation,
- assigning of j to the current block does not violate the cycle time constraint: $t_w(V_k \setminus V_{kl}) + t_b(V_{kl} \cup \{j\}) \le T_0$,
- after assigning j to the current block, the following inequality must hold: $\varphi(k) > \rho$, if $(V_k \cup \{j\}) \cap \widetilde{V} \ne \emptyset$.

The last condition assures that the feasible solution obtained will have $\mathcal{F}$-stability radius greater than ρ.

If $\mathcal{CL}(k,l) = \emptyset$, no more tasks can be assigned to the current block. A new block is opened (if $l < b_0$) and $\mathcal{CL}(k,l+1)$ is built, otherwise a new workstation with one empty block is opened and $\mathcal{CL}(k+1,1)$ is built. If $\mathcal{CL}(k,l) \ne \emptyset$, a task j is randomly chosen from $\mathcal{CL}(k,l)$ and assigned to V_{kl}, then $\mathcal{CL}(k,l)$ is rebuilt.

6.3. Computational experiments

The suggested algorithm was developed in C++ and evaluated on 2 series of 20 benchmarks close to real industrial problems. These benchmarks were generated taking into account real-life data of typical parts shapes manufactured in

Table 1
Benchmark tests.

	Series 1					Series 2				
	#V	T_0	OS	$\#E^b$	$\#E^m$	#V	T_0	OS	$\#E^b$	$\#E^m$
1	68	5.15	43.6	940	734	99	7.05	41.3	2210	1576
2	71	5.5875	51	1029	749	111	4.0125	45.7	2656	2230
3	78	4.6125	38.7	1399	1026	94	6.1375	44.3	1754	1662
4	71	6.575	53	1127	694	122	5.0625	43.8	2879	2549
5	72	7.8975	42.1	926	1018	105	4.9	47.8	1918	2318
6	74	5.5375	49.8	1266	705	97	5.4375	45.9	2128	1610
7	71	4.5875	49.7	998	793	101	4.3	36.9	2117	1566
8	75	6.0125	47.1	1071	911	87	5.7625	32.9	1427	1364
9	75	7.8975	49.5	1098	1009	110	5.475	40.3	2379	2073
10	76	5.7	41.1	1259	923	113	4.375	45.9	2784	2281
11	81	4.275	42.8	1567	918	113	5.85	39.5	2492	2370
12	92	5.9	46.5	1787	1203	96	4.7125	49.5	1815	1431
13	71	4.8875	52.7	1131	786	116	4.375	47.3	2332	2707
14	65	2.9125	42.8	1043	574	121	5.075	44.7	2699	2339
15	46	3.925	40	465	302	111	5.1375	39.7	2402	2008
16	74	5.075	42	1089	912	98	5.075	35.8	2218	1602
17	74	5.2	45.8	1039	1142	114	4.9875	48.2	2408	2421
18	70	3.9375	38.6	990	909	99	5.975	44	1856	1653
19	69	3.85	44.5	939	767	119	4.625	44.5	2615	2706
20	64	6.1125	38.9	770	783	127	7.8975	40.7	4034	2245

Table 2
Results for Series 1.

	10%				20%				30%			
	$\mathcal{N}$	$\mathcal{Z}_{\text{av}}$	ρ_{av}	$\mathcal{Q}$	$\mathcal{N}$	$\mathcal{Z}_{\text{av}}$	ρ_{av}	$\mathcal{Q}$	$\#\mathcal{S}$	$\mathcal{Z}_{\text{av}}$	ρ_{av}	$\mathcal{Q}$
1	8	27.25	3.17	20.34	12	29.21	2.51	26.08	3	26.67	1.35	19.07
2	7	31.21	2.74	14.86	13	33.58	2.30	18.17	5	31.10	1.38	14.49
3	2	47.00	2.25	5.81	5	47.70	2.03	15.80	13	49.50	1.95	19.82
4	7	27.00	4.25	6.03	9	29.00	3.45	25.75	3	26.50	1.52	25.73
5	7	35.00	4.12	17.58	5	34.70	2.50	22.20	4	34.13	1.29	16.40
6	3	35.50	1.51	140.97	7	36.43	1.63	82.42	10	38.10	1.63	95.61
7	7	27.43	2.10	18.40	4	26.50	1.30	11.88	8	27.00	1.17	19.84
8	4	25.75	3.31	13.68	8	27.44	2.48	18.65	12	28.71	2.12	12.71
9	8	42.25	3.96	48.16	3	41.00	1.54	48.42	4	41.25	1.52	71.79
10	7	37.64	3.33	25.72	13	39.50	2.95	46.94	7	36.50	1.32	38.45
11	5	44.70	2.01	18.42	11	45.77	1.48	16.19	11	45.64	1.22	39.21
12	7	44.36	3.89	53.32	16	46.81	2.81	20.42	26	49.62	2.34	42.24
13	2	36.50	1.16	85.17	4	37.50	1.27	92.99	6	38.50	1.25	53.89
14	2	36.00	1.52	12.50	6	38.00	1.27	19.58	13	40.35	1.26	16.04
15	5	28.70	2.46	22.50	5	29.00	1.56	20.36	7	30.00	1.47	51.42
16	3	42.50	2.01	18.00	6	43.50	1.47	27.93	5	43.80	1.45	19.77
17	4	41.13	1.15	96.90	7	42.50	1.19	43.72	9	42.89	1.12	444.01
18	3	41.50	1.42	11.04	8	42.81	1.11	144.58	9	43.50	1.05	57.82
19	4	30.00	1.12	44.15	6	30.58	1.05	127.30	8	31.56	1.02	62.15
20	5	21.80	2.68	30.94	6	22.25	2.11	21.08	12	23.25	1.78	94.90

transfer lines equipped with unit-built machines. Due to this, they represent relatively well real cases. The input data of these benchmarks can be found at http://www.emse.fr/~gurevsky/GALBP.zip and are presented in Table 1, where

- # is the cardinality of the corresponding set,
- $OS = \frac{2|\mathcal{A}|}{|V|(|V|-1)} \cdot 100\%$ is the order strength of the precedence constraints represented by graph $G = (V, \mathcal{A})$.

Series 2 in comparison with Series 1 is characterized by greater number of tasks and exclusion constraints. In addition, for these two series the following input data was used: $C_1 = 1$, $C_2 = 0.5$, $b_0 = 4$, $T_{\max} = 25$ s. The experiments were carried out on Intel Celeron 550 (2 GHz, 1 GB RAM).

For each benchmark, nine tests were conducted with the percentage of uncertain tasks sequentially chosen from the following set $\{10\%, 20\%, \ldots, 90\%\}$. Each new test was obtained from the previous keeping the same uncertain tasks and generating 10% new ones.

Tables 2–7 show the results obtained with the suggested algorithm for Series 1 and 2, respectively. The following notations are used:

- $\mathcal{N}$ is the number of found non-dominated solutions (i.e. $\mathcal{N} = \#\mathcal{AND}\mathcal{S}$),
- $\mathcal{Z}_{\text{av}}$ is the average value of the line cost for $\mathcal{AND}\mathcal{S}$;
- ρ_{av} is the average value of the $\mathcal{F}$-stability radius for $\mathcal{AND}\mathcal{S}$,

Table 3
Results for Series 1.

	40%				50%				60%			
	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$
1	7	27.79	1.36	21.96	11	29.27	1.35	30.61	13	29.31	1.23	12.36
2	8	31.50	1.24	57.60	14	33.25	1.33	25.86	14	34.00	1.38	22.34
3	9	48.94	1.44	16.10	12	49.79	1.28	14.59	16	50.91	1.24	23.48
4	6	27.50	1.53	31.06	6	27.58	1.47	30.43	9	28.39	1.54	29.28
5	5	34.70	1.27	30.61	7	35.29	1.25	22.77	10	36.00	1.16	25.04
6	15	39.67	1.63	82.43	21	40.60	1.65	167.47	22	41.50	1.58	75.77
7	7	26.50	1.04	37.60	10	27.65	1.14	32.31	11	28.18	1.09	32.31
8	6	27.00	1.62	8.89	9	27.22	1.57	9.00	12	27.92	1.40	19.14
9	4	41.88	1.70	33.03	9	42.83	1.57	89.83	9	43.33	1.64	24.47
10	10	37.70	1.48	18.18	12	38.58	1.40	23.31	14	39.25	1.40	21.06
11	13	46.96	1.20	27.09	11	47.86	1.19	19.95	14	47.82	1.02	33.77
12	31	51.68	2.25	31.38	40	53.85	2.23	78.31	19	47.76	1.31	30.17
13	8	39.50	1.23	47.38	12	40.79	1.19	72.91	16	42.38	1.24	61.42
14	18	42.11	1.20	23.65	21	42.88	1.02	28.91	22	45.20	1.01	23.45
15	11	32.00	1.52	45.64	13	33.00	1.41	49.67	21	34.90	1.40	41.95
16	10	44.40	1.28	36.25	12	45.04	1.25	35.40	14	46.29	1.28	32.32
17	13	44.00	1.15	51.53	15	45.07	1.16	32.46	18	45.81	1.12	28.29
18	15	44.53	0.96	60.58	18	45.42	0.92	42.47	22	47.11	0.95	53.50
19	10	32.20	1.05	134.33	18	34.53	1.07	371.72	19	35.29	1.03	171.61
20	14	23.75	1.86	60.13	13	25.19	2.02	101.44	16	25.41	1.93	94.77

Table 4
Results for Series 1.

	70%				80%				90%			
	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$
1	11	29.23	1.20	16.02	15	29.57	1.08	30.71	15	29.83	1.09	29.06
2	18	34.61	1.30	17.68	21	34.81	1.27	84.76	22	36.07	1.29	28.30
3	18	51.50	1.19	18.65	22	53.52	1.25	10.74	25	54.28	1.20	9.88
4	12	28.83	1.50	19.53	11	28.82	1.41	27.52	16	30.22	1.47	29.32
5	11	36.36	1.21	18.65	11	36.50	1.13	25.44	8	36.50	1.06	13.28
6	25	43.16	1.69	31.21	30	44.22	1.54	121.62	32	45.73	1.57	249.80
7	13	28.73	1.14	17.96	13	29.46	1.18	16.45	11	28.32	1.04	16.63
8	14	29.25	1.51	11.56	14	29.21	1.48	8.70	13	28.92	1.38	12.95
9	11	43.45	1.46	33.81	13	43.73	1.26	97.48	12	44.79	1.43	46.64
10	14	39.68	1.37	15.82	19	39.58	1.26	22.78	20	40.93	1.31	21.35
11	22	50.25	1.07	27.68	28	52.16	1.03	1856.39	33	53.81	1.04	19.23
12	22	48.80	1.31	43.07	26	50.12	1.35	20.39	25	49.88	1.26	32.84
13	22	42.89	1.14	54.37	21	44.00	1.16	45.26	18	44.50	1.16	48.91
14	26	46.10	0.99	22.65	31	47.58	0.95	22.14	34	48.75	0.89	19.83
15	19	36.00	1.38	52.67	17	35.18	1.26	49.86	21	35.14	1.13	36.89
16	18	46.83	1.23	19.00	24	48.25	1.18	16.80	22	48.70	1.19	14.89
17	21	47.67	1.22	21.66	22	47.70	1.12	50.87	31	48.58	1.10	624.87
18	26	48.42	0.97	38.43	29	49.26	0.91	57.88	28	50.71	0.93	42.47
19	17	35.47	1.03	192.54	14	35.36	0.99	192.54	15	35.50	0.98	192.54
20	19	25.53	1.94	94.41	19	27.05	2.02	94.77	15	25.13	1.45	92.04

- $\mathcal{Q}$ is a specific measure of the constructed front $\mathcal{ANDS}$ that analyzes the situation where the relative augmentation of the $\mathcal{F}$-stability radius is most important with respect to the relative augmentation of the line cost between any two neighbor solutions of $\mathcal{ANDS}$. In other words,

$$\mathcal{Q} = \max_{i \in \mathcal{I}_{\mathcal{N}-1}} \left\{ \frac{\rho_{\mathcal{F}}(s_{i+1}, t) - \rho_{\mathcal{F}}(s_i, t)}{\rho_{\mathcal{F}}(s_i, t)} \bigg/ \frac{\mathcal{Z}(s_{i+1}, t) - \mathcal{Z}(s_i, t)}{\mathcal{Z}(s_i, t)} \right\},$$

  under the condition that non-dominated solutions $\mathcal{ANDS} = \{s_1, s_2, \dots, s_{\mathcal{N}}\}$ are sorted in increasing order with regard to their values of the line cost (or the $\mathcal{F}$-stability radius). In some respect, the value of $\mathcal{Q}$ characterizes the quality of the compromise provided by solutions of $\mathcal{ANDS}$.

Analyzing the results obtained, it can be concluded that the algorithm suggested has provided in average a greater number of non-dominated solutions for Series 2 than for Series 1 with respect to the percentage of uncertain tasks (see Fig. 5). However, the opposite conclusion can be made for the average value of ρ_{av} (see Fig. 6). It can be explained by the fact that increasing the number of tasks and constraints leads to increasing the number of feasible solutions constructed in each stage of the algorithm. As a consequence, following the logic of the proposed approach, this implies the augmentation of the distribution density of solutions in $\mathcal{ANDS}$. An illustrative example is given in Fig. 7, where two fronts with the same percentage of uncertain tasks and the same number of non-dominated solutions for Series 1 and 2 are shown.

Table 5
Results for Series 2.

	10%				20%				30%			
	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$	$\#\mathcal{S}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$
1	4	42.00	5.08	13.70	13	44.65	3.82	27.22	21	46.07	3.33	19.06
2	4	62.88	1.29	63.12	9	64.50	1.08	25.04	17	66.79	0.99	61.00
3	5	49.00	2.27	50.38	9	50.17	2.22	35.23	12	50.75	2.03	26.75
4	6	57.50	1.59	129.55	8	57.25	0.90	826.07	14	58.82	1.03	74.65
5	5	48.30	1.68	50.07	11	48.77	1.43	291.70	11	49.32	1.15	29.72
6	7	48.71	1.28	74.35	13	50.23	1.36	41.88	15	51.53	1.38	71.91
7	8	42.69	1.71	54.35	10	43.30	1.06	61.64	13	44.58	0.95	71.73
8	3	45.50	2.76	7.86	3	46.00	2.53	17.50	13	48.69	2.42	16.29
9	6	36.75	3.52	29.24	4	35.63	1.44	50.65	5	36.40	1.51	39.70
10	6	63.75	1.50	46.04	9	64.94	1.27	43.39	13	66.08	1.12	52.16
11	4	35.38	1.18	145.01	8	36.81	1.38	92.11	11	37.50	1.30	44.29
12	3	33.50	0.96	121.28	5	33.90	1.13	374.57	9	35.00	1.07	111.54
13	11	48.64	2.43	529.05	9	47.11	1.14	32.72	7	48.07	1.22	30.91
14	5	43.80	1.35	195.40	8	45.13	1.24	36.27	13	45.85	1.05	49.56
15	3	36.00	0.76	246.17	6	37.42	1.21	79.48	13	39.04	1.20	383.44
16	5	45.00	2.93	15.41	5	45.00	1.48	51.21	11	46.50	1.35	38.29
17	9	50.72	3.02	263.52	1	48.50	2.10	0.00	8	50.44	1.56	74.23
18	3	38.00	2.12	42.27	8	39.50	1.83	47.31	13	40.81	1.68	56.87
19	5	41.70	1.35	41.18	9	42.94	1.21	45.96	15	44.27	1.04	24.48
20	2	62.50	0.60	86.36	2	62.50	0.64	58.47	2	62.50	0.55	145.92

Table 6
Results for Series 2.

	40%				50%				60%			
	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$
1	16	44.94	2.73	10.55	23	46.67	2.52	9.52	12	43.71	1.60	38.40
2	27	68.78	0.96	75.74	29	70.07	0.92	53.75	32	72.70	0.90	55.31
3	16	51.88	2.11	22.22	25	55.36	2.10	27.10	15	52.37	1.46	23.46
4	16	61.25	1.16	66.29	23	61.67	1.10	263.00	25	63.78	1.14	48.33
5	15	50.50	1.14	568.28	17	51.26	1.12	322.58	24	52.48	1.03	34.36
6	19	52.42	1.15	58.32	22	52.43	1.03	736.17	24	54.42	1.14	48.88
7	19	46.16	0.91	157.69	22	47.59	0.93	68.32	25	47.76	0.82	42.14
8	18	50.75	2.28	26.76	26	54.17	2.23	24.76	14	49.00	1.36	83.44
9	12	37.92	1.40	23.77	14	38.96	1.38	54.90	17	39.53	1.35	54.90
10	18	66.97	1.02	79.45	25	69.94	1.05	108.55	32	71.56	1.00	57.13
11	10	38.25	1.34	102.55	13	39.35	1.35	28.21	15	39.13	1.18	197.48
12	13	36.23	0.99	309.72	13	36.77	0.99	148.97	17	37.65	0.96	93.43
13	15	49.13	1.01	83.65	16	49.50	0.98	52.29	20	51.15	0.93	167.56
14	13	46.65	1.05	95.99	17	47.41	0.96	125.24	23	48.67	0.91	161.70
15	14	39.75	1.22	35.38	15	42.07	1.31	29.38	20	41.78	1.12	177.42
16	16	47.84	1.22	32.24	18	48.44	1.17	36.27	17	48.41	1.08	34.41
17	16	53.31	1.62	19.67	15	54.70	1.61	34.84	30	56.72	1.47	606.51
18	12	40.38	1.30	24.43	14	40.79	1.30	30.97	15	41.40	1.26	31.11
19	17	45.94	1.07	30.46	16	45.84	1.01	31.01	25	47.28	1.02	30.07
20	4	62.75	0.63	154.98	4	62.88	0.61	234.76	3	63.00	0.61	195.25

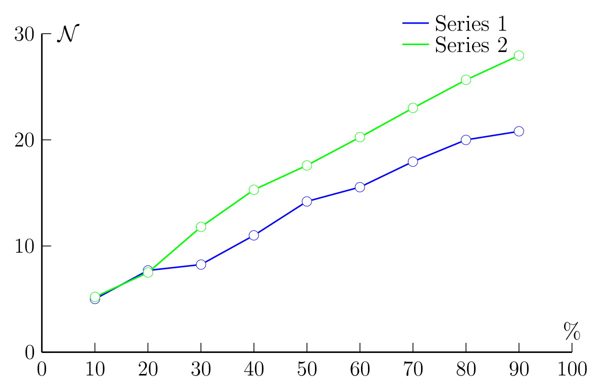

Fig. 5. Average value of $\mathcal{N}$ with respect to the percentage of uncertain tasks.

Table 7
Results for Series 2.

	70%				80%				90%			
	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$	$\mathcal{N}$	$\mathcal{Z}_{av}$	ρ_{av}	$\mathcal{Q}$
1	17	44.53	1.65	24.22	17	45.18	1.62	22.44	20	45.85	1.60	26.35
2	37	74.53	0.92	78.25	47	77.41	0.91	44.24	46	79.33	0.92	40.36
3	19	52.89	1.39	32.02	25	54.84	1.38	60.01	25	55.42	1.33	38.20
4	28	63.18	0.98	165.45	32	65.08	1.03	1094.33	41	67.26	1.04	294.15
5	23	53.87	1.07	40.21	25	54.14	1.01	49.18	32	55.97	1.04	37.45
6	29	55.41	1.13	52.42	30	56.50	1.14	47.00	30	56.35	1.15	71.53
7	30	49.50	0.94	183.32	31	49.29	0.87	147.55	34	51.28	0.93	147.71
8	16	49.31	1.28	67.31	19	50.42	1.31	45.17	21	52.02	1.38	44.74
9	19	40.50	1.35	54.90	19	39.95	1.33	54.90	17	41.18	1.39	54.90
10	38	74.07	1.04	68.07	42	73.92	0.98	69.89	42	75.99	1.01	75.73
11	23	40.78	1.20	58.82	19	40.87	1.18	93.23	22	41.75	1.18	41.62
12	19	39.03	1.03	100.58	26	39.90	1.03	39.90	22	39.18	0.95	176.77
13	21	52.40	0.92	48.32	21	53.40	1.00	24.50	29	52.95	0.86	156.31
14	24	50.69	1.07	46.24	32	51.95	1.03	128.34	33	51.89	1.03	165.25
15	22	42.77	1.14	70.56	22	43.52	1.18	111.08	27	43.50	1.11	213.10
16	22	49.80	1.09	41.78	27	51.44	1.17	28.49	28	52.14	1.13	61.46
17	28	57.91	1.47	44.05	31	59.55	1.45	91.69	41	61.01	1.41	101.16
18	19	42.03	1.31	33.51	17	42.38	1.25	37.84	20	43.48	1.30	29.99
19	21	47.24	0.99	38.67	25	48.18	0.99	34.55	23	48.04	1.02	24.07
20	5	63.30	0.65	97.78	6	63.67	0.58	132.86	6	63.67	0.61	97.68

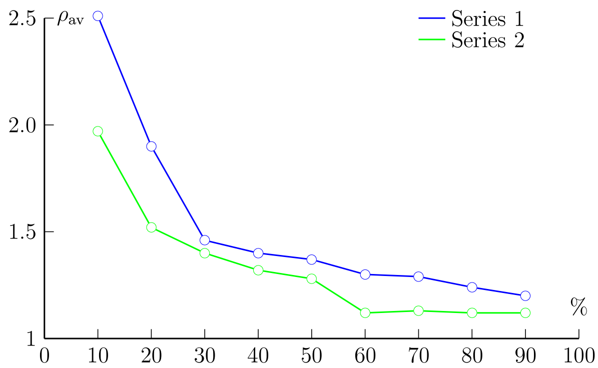

Fig. 6. Average value of ρ_{av} with respect to the percentage of uncertain tasks.

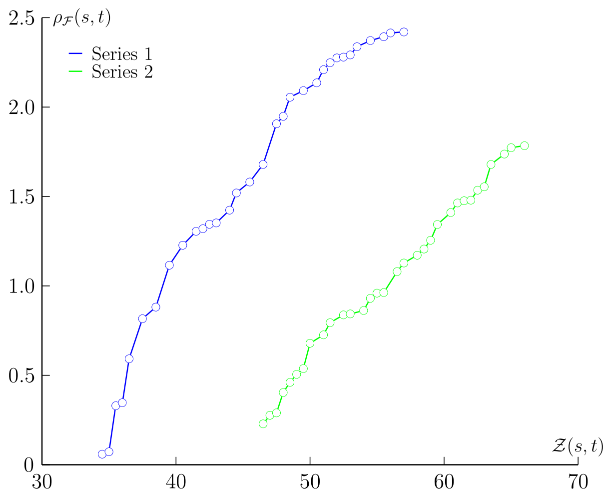

Fig. 7. $\mathcal{AND}\mathcal{S}$ for Test 6 of Series 1 and for Test 5 of Series 2 with 90% of uncertain tasks.

The average value of $\mathcal{Q}$ is 60.91 for Series 1 and 100.27 for Series 2. These rather great values of Q show the importance of the location of alternative solutions.

7. Conclusions

Production lines with workstations having sequential workplaces where tasks are executed in parallel were considered. The line balancing problem for these lines was formulated and studied under variations of task processing times. Conditions of stability and the algorithms to calculate the stability radii for feasible, quasi-feasible, and optimal solutions were investigated. Polynomial time algorithms were suggested for the case of feasible and quasi-feasible solutions.

Note that the results obtained can be used for other line balancing problems having more simplified assumptions, for example SALBP-1.

Taking into account the stability measure suggested in this paper, the initial optimization problem becomes multi-objective aiming to optimize the line cost and its robustness simultaneously. Since these two objectives are contradictory, the concept of Pareto-optimality was used. A multi-stage approximate algorithm to find an approximation of Pareto front was developed and evaluated on benchmarks.

The results obtained offer a possibility for decision makers to include a robustness measure in their design process and to evaluate line configurations not only in terms of their cost but also regarding their stability under small variations of task processing times. The future work will concern the stability analysis for the design process for lines having different layouts and technological constraints.

Acknowledgment

The authors thank Chris Yukna for his help with English.

References

[1] K. Ağpak, H. Gökçen, A chance-constrained approach to stochastic line balancing problem, European Journal of Operational Research 180 (3) (2007) 1098–1115.
[2] O. Battaïa, A. Dolgui, Reduction approaches for a generalized line balancing problem, Computers & Operations Research 39 (10) (2012) 2337–2345.
[3] A. Baykasoğlu, L. Özbakır, Stochastic *U*-line balancing using genetic algorithms, The International Journal of Advanced Manufacturing Technology 32 (1–2) (2007) 139–147.
[4] T. Belgacem, M. Hifi, Sensitivity analysis of the knapsack problem: tighter lower and upper bound limits, Journal of Systems Science and Systems Engineering 17 (2) (2008) 156–170.
[5] T. Belgacem, M. Hifi, Sensitivity analysis of the knapsack sharing problem: perturbation of the profit of an item, International Transactions in Operational Research 15 (1) (2008) 35–49.
[6] T. Belgacem, M. Hifi, Sensitivity analysis of the knapsack sharing problem: perturbation of the weight of an item, Computers & Industrial Engineering 35 (1) (2008) 295–308.
[7] T. Belgacem, M. Hifi, Sensitivity analysis of the optimum to perturbation of the profit of a subset of items in the binary knapsack problem, Discrete Optimization 5 (4) (2008) 755–761.
[8] H. Bräsel, Y. Sotskov, F. Werner, Stability of a schedule minimizing mean flow time, Mathematical and Computer Modelling 24 (10) (1996) 39–53.
[9] W.-C. Chiang, T. Urban, The stochastic *U*-line balancing problem: a heuristic procedure, European Journal of Operational Research 175 (3) (2006) 1767–1781.
[10] A. Dolgui, B. Finel, O. Guschinskaya, N. Guschinsky, G. Levin, F. Vernadat, Balancing large-scale machining lines with multi-spindle heads using decomposition, International Journal of Production Research 44 (18–19) (2006) 4105–4120.
[11] M. Ehrgott, Multicriteria Optimization, second ed., Springer, Berlin/Heidelberg, 2005.
[12] E. Erel, I. Sabuncuoglu, H. Sekerci, Stochastic assembly line balancing using beam search, International Journal of Production Research 43 (7) (2005) 1411–1426.
[13] R. Gamberini, E. Gebennini, A. Grassi, A. Regattieri, A multiple single-pass heuristic algorithm solving the stochastic assembly line rebalancing problem, International Journal of Production Research 47 (8) (2009) 2141–2164.
[14] M. Gen, Y. Tsujimura, Y. Li, Fuzzy assembly line balancing using genetic algorithms, Computers & Industrial Engineering 31 (3–4) (1996) 631–634.
[15] O. Guschinskaya, A. Dolgui, Comparison of exact and heuristic methods for a transfer line balancing problem, International Journal of Production Economics 120 (2) (2009) 276–286.
[16] O. Guschinskaya, A. Dolgui, N. Guschinsky, G. Levin, A heuristic multi-start decomposition approach for optimal design of serial machining lines, European Journal of Operational Research 189 (3) (2008) 902–913.
[17] O. Guschinskaya, E. Gurevsky, A. Dolgui, A. Eremeev, Metaheuristic approaches for the design of machining lines, The International Journal of Advanced Manufacturing Technology 55 (1–4) (2011) 11–22.
[18] M. Hifi, H. Mhalla, S. Sadfi, Sensitivity of the optimum to perturbations of the profit or weight of an item in the binary knapsack problem, Journal of Combinatorial Optimization 10 (3) (2005) 239–260.
[19] M. Hifi, H. Mhalla, S. Sadfi, An adaptive algorithm for the knapsack problem: perturbation of the profit or weight of an arbitrary item, European Journal of Industrial Engineering 2 (2) (2008) 134–152.
[20] N. Hop, A heuristic solution for fuzzy mixed-model line balancing problem, European Journal of Operational Research 168 (3) (2006) 798–810.
[21] S. Kravchenko, Y. Sotskov, F. Werner, Optimal schedules with infinitely large stability radius, Optimization 33 (3) (1995) 271–280.
[22] M. Libura, E. van der Poort, G. Sierksma, J. van der Veen, Stability aspects of the traveling salesman problem based on *k*-best solutions, Discrete Applied Mathematics 87 (1–3) (1998) 159–185.
[23] S. Liu, H. Ong, H. Huang, A bidirectional heuristic for stochastic assembly line balancing type II problem, The International Journal of Advanced Manufacturing Technology 25 (1–2) (2005) 71–77.
[24] R. Ramaswamy, J. Orlin, N. Chakravarti, Sensitivity analysis for shortest path problems and maximum capacity path problems in undirected graphs, Mathematical Programming 102 (2) (2005) 355–369.
[25] B. Rekiek, A. Dolgui, A. Delchambre, A. Bratcu, State of art of optimization methods for assembly line design, Annual Reviews in Control 26 (2) (2002) 163–174.
[26] A. Scholl, Balancing and Sequencing of Assembly Lines, second ed., Physica-Verlag, Heidelberg, 1999.
[27] Y. Sotskov, Stability of an optimal schedule, European Journal of Operational Research 55 (1) (1991) 91–102.
[28] Y. Sotskov, A. Dolgui, M.-C. Portmann, Stability analysis of an optimal balance for an assembly line with fixed cycle time, European Journal of Operational Research 168 (3) (2006) 783–797.
[29] Y. Sotskov, N. Sotskova, T.-C. Lai, F. Werner, Scheduling Under Uncertainty: Theory and Algorithms, Belorusskaya Nauka, Minsk, 2010.
[30] Y. Sotskov, N. Sotskova, F. Werner, Stability of an optimal schedule in a job shop, Omega 25 (4) (1997) 397–414.

[31] Y. Sotskov, V. Tanaev, F. Werner, Stability radius of an optimal schedule: a survey and recent developments, in: G. Yu (Ed.), Industrial Applications of Combinatorial Optimization, in: Applied Optimization, vol. 16, Kluwer Academic Publishers, 1998, pp. 72–108.
[32] Y. Sotskov, A. Wagelmans, F. Werner, On the calculation of the stability radius of an optimal or an approximate schedule, Annals of Operations Research 83 (0) (1998) 213–252.
[33] S. Tasan, S. Tunali, A review of the current applications of genetic algorithms in assembly line balancing, Journal of Intelligent Manufacturing 19 (1) (2008) 49–69.
[34] Y. Tsujimura, M. Gen, E. Kubota, Solving fuzzy assembly-line balancing problem with genetic algorithms, Computers & Industrial Engineering 29 (1–4) (1995) 543–547.
[35] T. Urban, W.-C. Chiang, An optimal piecewise-linear program for the U-line balancing problem with stochastic task times, European Journal of Operational Research 168 (3) (2006) 771–782.
[36] E. van der Poort, 1997, Aspects of sensitivity analysis for the traveling salesman problem, Ph.D. Thesis, University of Groningen, Groningen, The Netherlands.